Adrian Ottewill

University College Dublin

LSC Meeting, Livingston LIGO Laboratory,
16–18 March 2000

- Environmental Correlations

- Multitaper methods

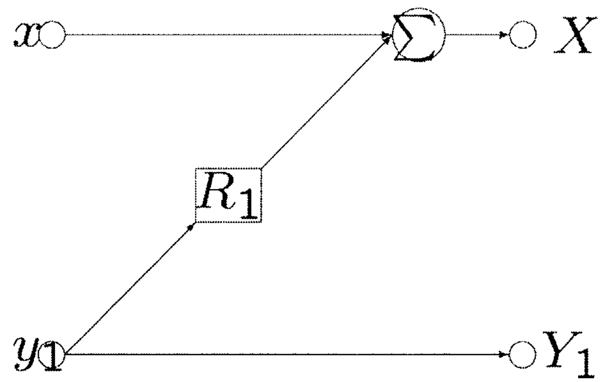
- Incorporating Grasp code into the DMT

LIGO-G000086-00-D

ENVIRONMENTAL CORRELATIONS

Automatic cross-talk removal from multi-channel data

Bruce Allen, Wensheng Hua and Adrian Ottewill



To estimate the transfer function requires an averaging process. which we carry out in frequency space:

$$\tilde{\mathbf{X}} = \left[\tilde{X}(1), \tilde{X}(2), \cdots, \tilde{X}(M/2)\right] = \left[\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)}, \cdots, \tilde{\mathbf{X}}^{(B)}\right]$$

so we have $F = (M/2)/B$ frequency ‘bins’ in each frequency band.

Given the transfer function $r^{(b)}$ within the frequency band, our estimate of the Fourier transform of the “true" channel of interest is

$$\tilde{\tilde{\mathbf{x}}}^{(b)} = \tilde{\mathbf{X}}^{(b)} - \sum_{i=1}^{N} r_i^{(b)} \tilde{\mathbf{Y}}_i^{(b)}.$$

We assume that the best estimate of the transfer function in the frequency band b is the one that minimizes the norm

$$\left(\tilde{\tilde{\mathbf{x}}}^{(b)}, \tilde{\tilde{\mathbf{x}}}^{(b)}\right).$$

SOLUTION: Introduce correlation matrix estimate in the b'th band:

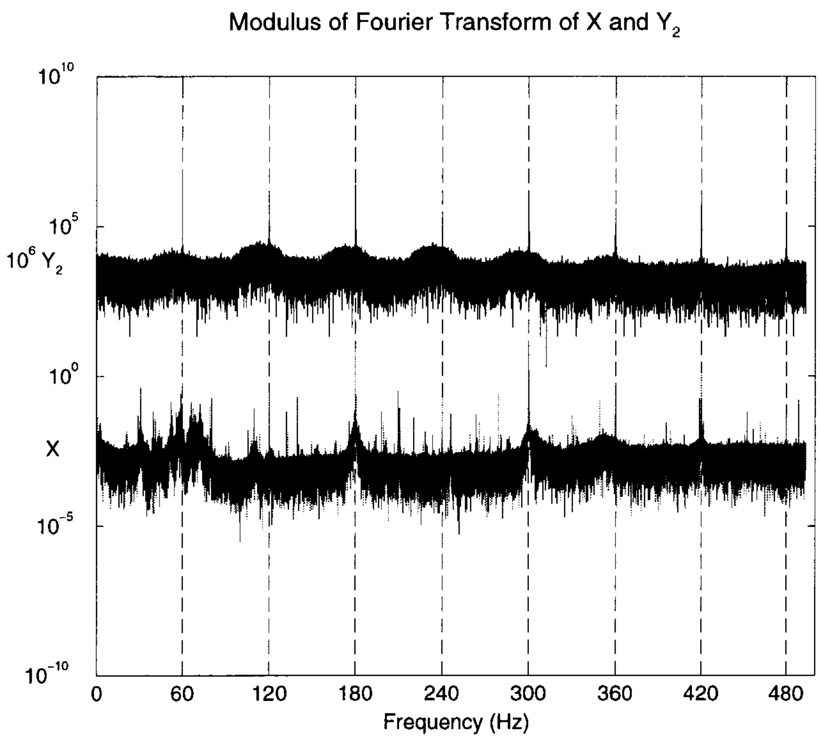
$$C_{ij}^{(b)} = \left(\tilde{\mathbf{Y}}_i^{(b)}, \tilde{\mathbf{Y}}_j^{(b)}\right).$$

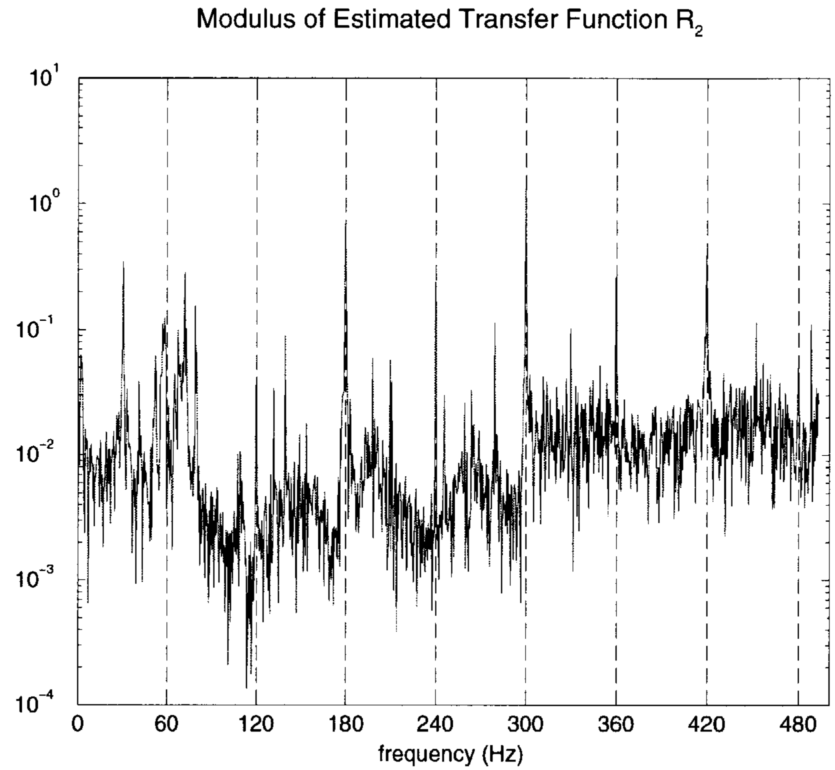
then

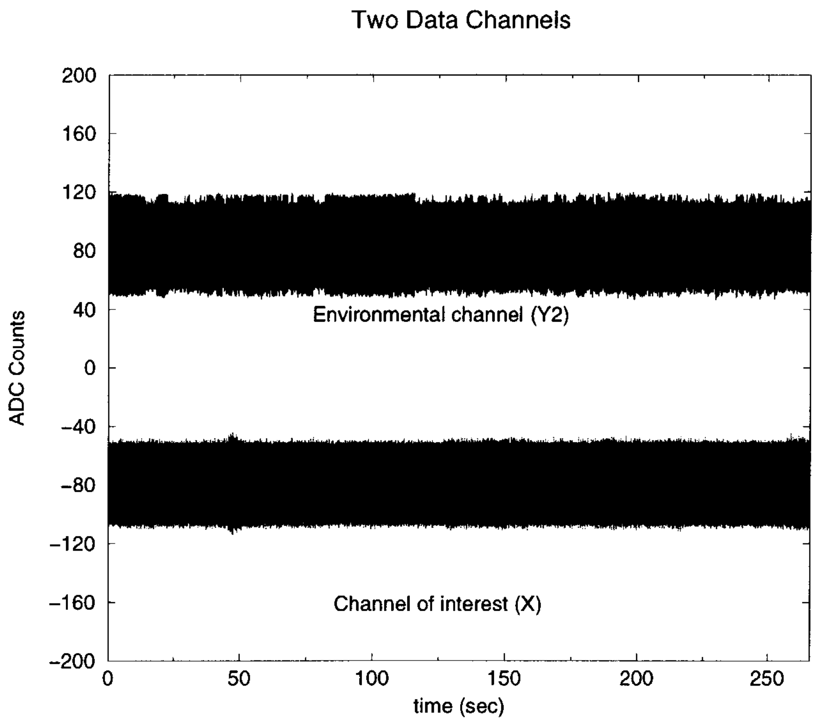
$$r_i^{(b)} = \sum_{j=1}^{N} \left(C^{-1}\right)_{ij} \left(\tilde{\mathbf{Y}}j^{(b)}, \tilde{\mathbf{X}}^{(b)}\right) \quad \text{for } i = 1, \cdots, N.$$

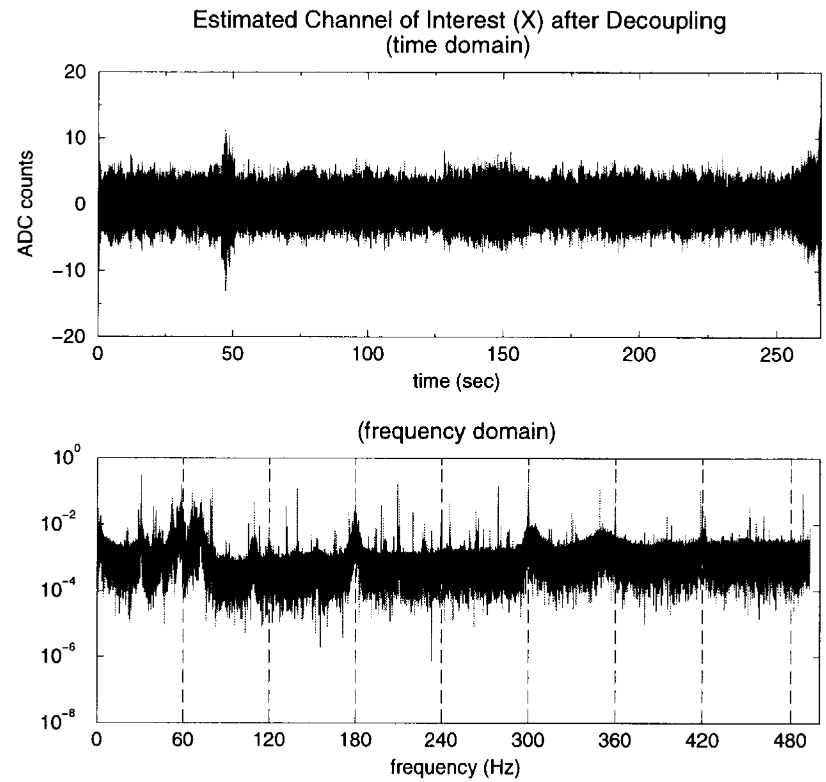
Also we can introduce as measure of the correlations:

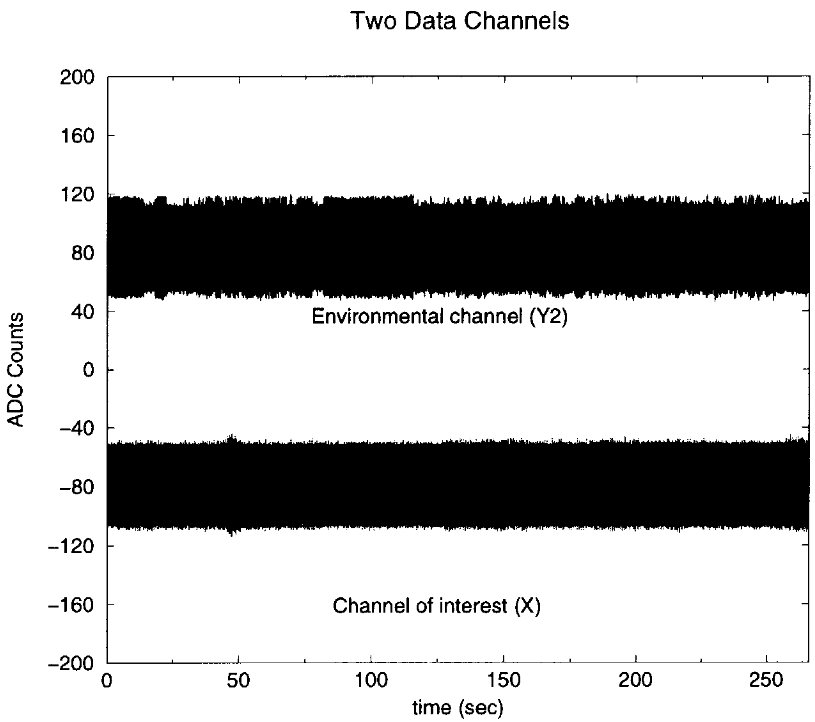
$$\left(\tilde{\tilde{\mathbf{x}}}^{(b)}, \tilde{\tilde{\mathbf{x}}}^{(b)}\right) = \left(\tilde{\mathbf{X}}^{(b)}, \tilde{\mathbf{X}}^{(b)}\right) \left[1 - |\rho^{(b)}|^2\right].$$

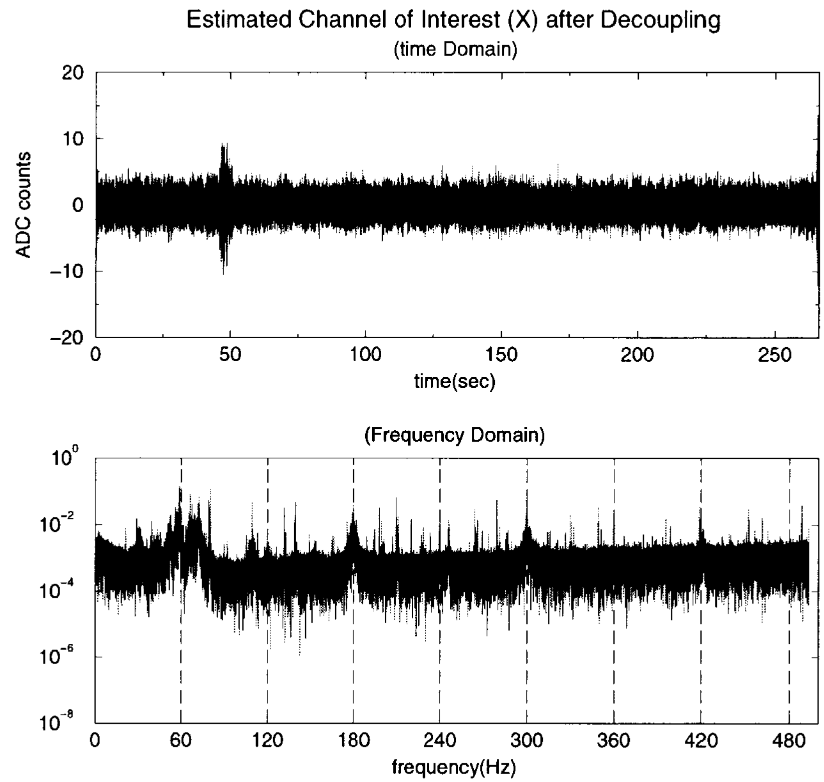

Modulus of Fourier Transform of X and Y_2
10^{10}
10^5
10^6 Y_2
10^0
X
10^{-5}
10^{-10}
0
60
120
180
240
300
360
420
480
Frequency (Hz)

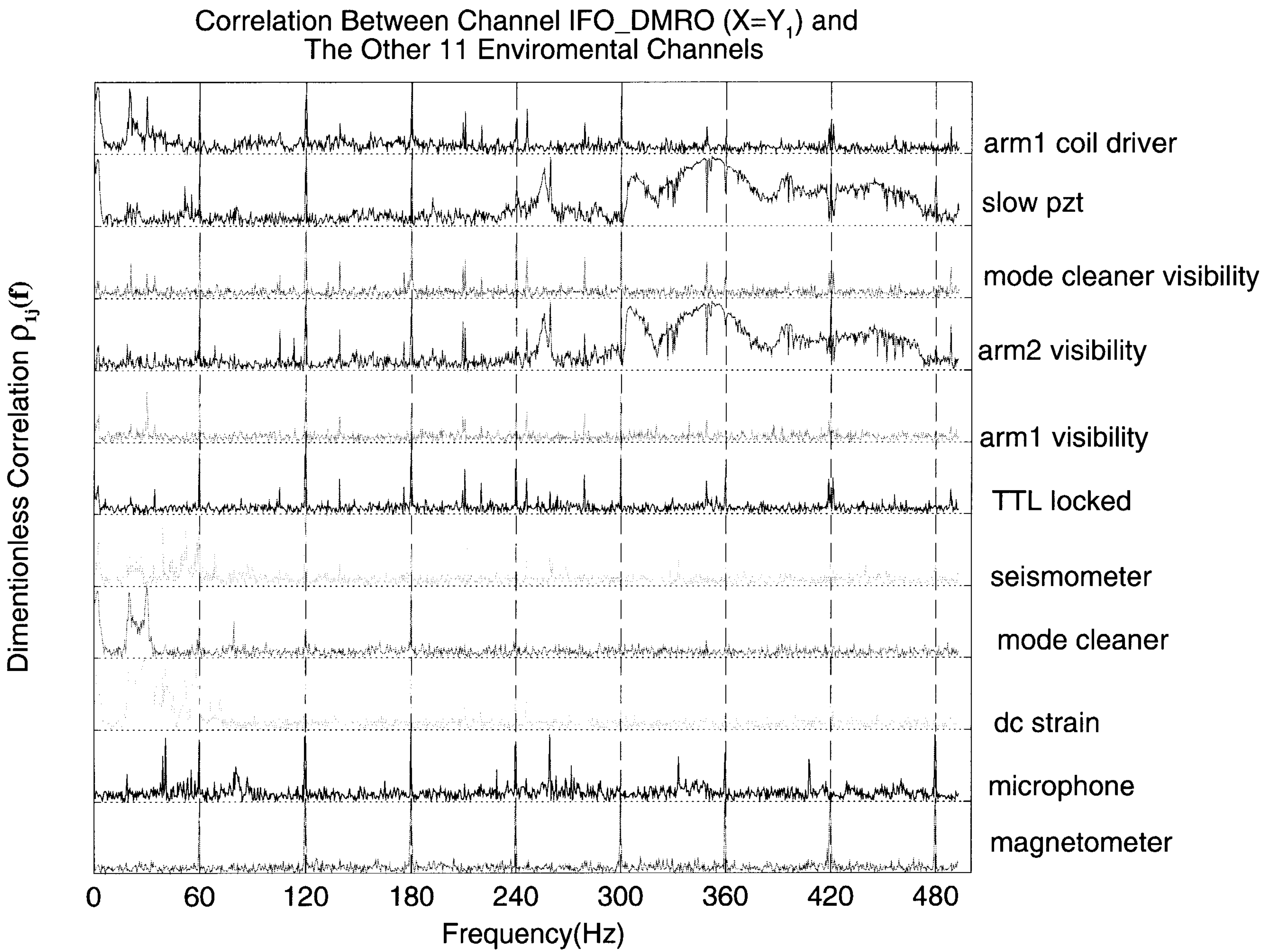

Modulus of Estimated Transfer Function R_2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
60
120
180
240
300
360
420
480
frequency (Hz)


Two Data Channels
ADC Counts
Environmental channel (Y2)
Channel of interest (X)
time (sec)


Estimated Channel of Interest (X) after Decoupling
(time domain)
ADC counts
time (sec)
(frequency domain)
frequency (Hz)


Two Data Channels
ADC Counts
200
160
120
80
40
0
−40
−80
−120
−160
−200
Environmental channel (Y2)
Channel of interest (X)
0
50
100
150
200
250
time (sec)


Estimated Channel of Interest (X) after Decoupling
(time Domain)
ADC counts
20
10
0
−10
−20
0
50
100
150
200
250
time(sec)
(Frequency Domain)
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
0
60
120
180
240
300
360
420
480
frequency(Hz)



Correlation Between Channel IFO_DMRO (X=Y1) and
The Other 11 Enviromental Channels
Dimentionless Correlation ρ1j(f)
arm1 coil driver
slow pzt
mode cleaner visibility
arm2 visibility
arm1 visibility
TTL locked
seismometer
mode cleaner
dc strain
microphone
magnetometer
0
60
120
180
240
300
360
420
480
Frequency(Hz)

MULTITAPER METHODS

Bruce Allen and Adrian Ottewill SPECTRAL ESTIMATORS

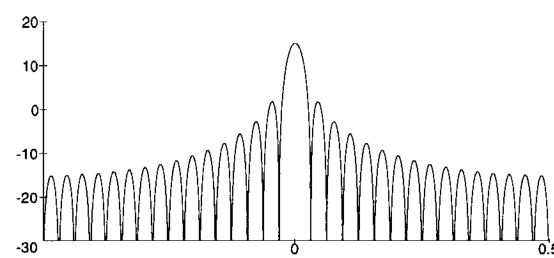
The Periodogram:

$$\widehat{S}_p^{(N)}(f) \equiv \frac{1}{N}\left|\tilde{X}^{(N)}(f)\right|^2 .$$

$$E\left(\widehat{S}_p^{(N)}(f)\right) = \int_{-f_{\rm ny}}^{f_{\rm ny}} \mathcal{F}^{(N)}(f-f')S(f)\,\mathrm{d}f,$$

where $\mathcal{F}_N(f)$ is the Fejér kernel

$$\mathcal{F}^{(N)}(f) = \frac{\Delta t}{N}\left|\sum_{r=1}^{N} e^{2\pi i f r \Delta t}\right|^2 .$$



$10\ \log_{10}(\mathcal{F}^{(N)}(f))$ for $N = 32$

Tapering (Windowing):

$$\widehat{S}_w^{(N)}(f) \equiv \Delta t \left| \sum_{r=1}^{N} w_r X_r e^{-2\pi i f r \Delta t} \right|^2, \qquad \sum_{r=1}^{N} {w_r}^2 = 1.$$

The modified periodogram is given by:

$$\widehat{S}_w^{(N)}(f) \equiv \Delta t \left| \sum_{r=1}^{N} w_r X_r e^{-2\pi i f r \Delta t} \right|^2 .$$

and correspondingly

$$E\left(\widehat{S}_w^{(N)}(f)\right) = \int_{-f_{\mathrm{ny}}}^{f_{\mathrm{ny}}} \mathcal{W}^{(N)}(f - f') S(f) \, \mathrm{d}f,$$

$$\mathcal{W}^{(N)}(f) = \Delta t \left| \sum_{r=1}^{N} w_r e^{2\pi i f r \Delta t} \right|^2 .$$

Slepian Tapers

These are constructed by first choosing a width W for the central lobe and then deman

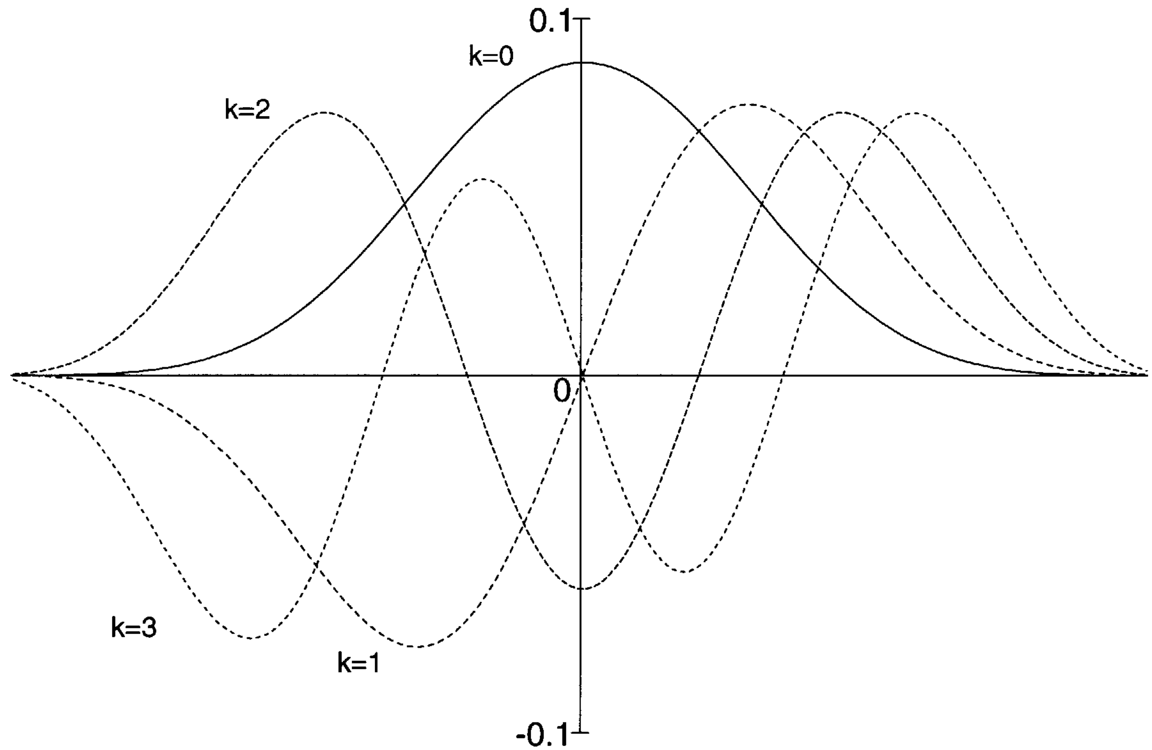
$$E_W \equiv \int_{-W}^{W} \left|\tilde{w}^{(N)}(f)\right|^2 \mathrm{d}f,$$

In the time domain this corresponds to extremising

$$\sum_{r=1}^{N} \sum_{s=1}^{N} w_r A_{rs} w_s \qquad \text{subject to} \qquad \sum_{r=1}^{N} w_r^2 = 1$$

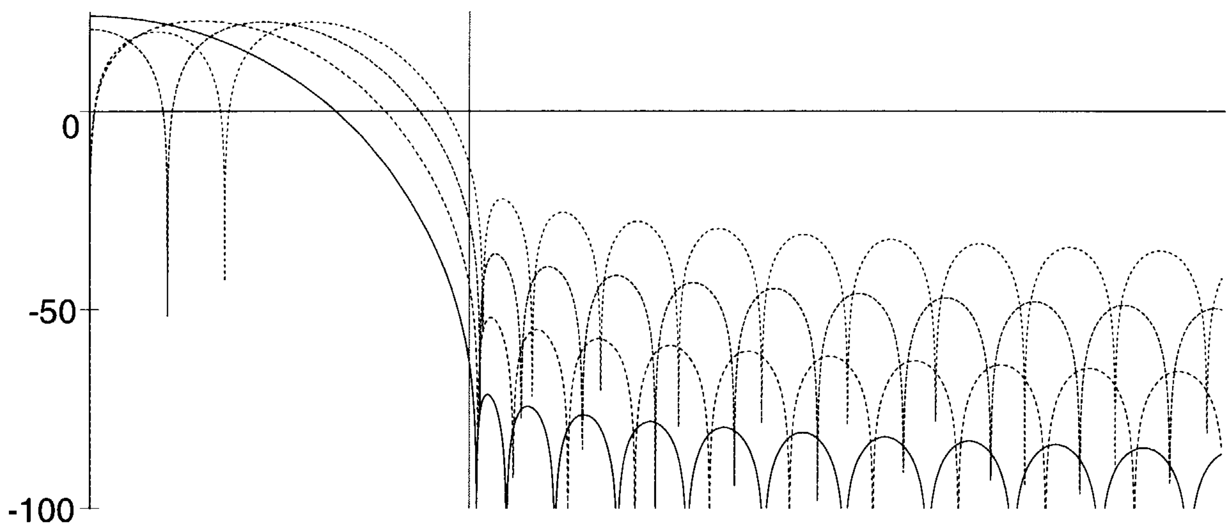
where

$$A_{rs} = \frac{\sin 2\pi W(r-s)}{\pi(r-s)} \qquad \text{real symmetric.}$$



The first four Slepian tapers for $N = 512$, $W = 4/(N\Delta t)$.

Eigenvalues $1 > \lambda^{(0)} > \lambda^{(1)} > \cdots > \lambda^{(N-1)} > 0$ measure energy confinement. First $2NW\Delta t - 1$ eigenvalues are close to one and so provide excellent energy confinement while thereafter they fall rapidly to zero.



The first four Slepian tapers measured by $10 \log_{10}(\mathcal{W}_k^{(N)}(f))$. The corresponding eigenvalues are $\lambda^{(0)} = 0.9999999997$ $\lambda^{(1)} = 0.9999999693$, $\lambda^{(2)} = 0.9999984555$, $\lambda^{(2)} = 0.9999482420$ (10DP).

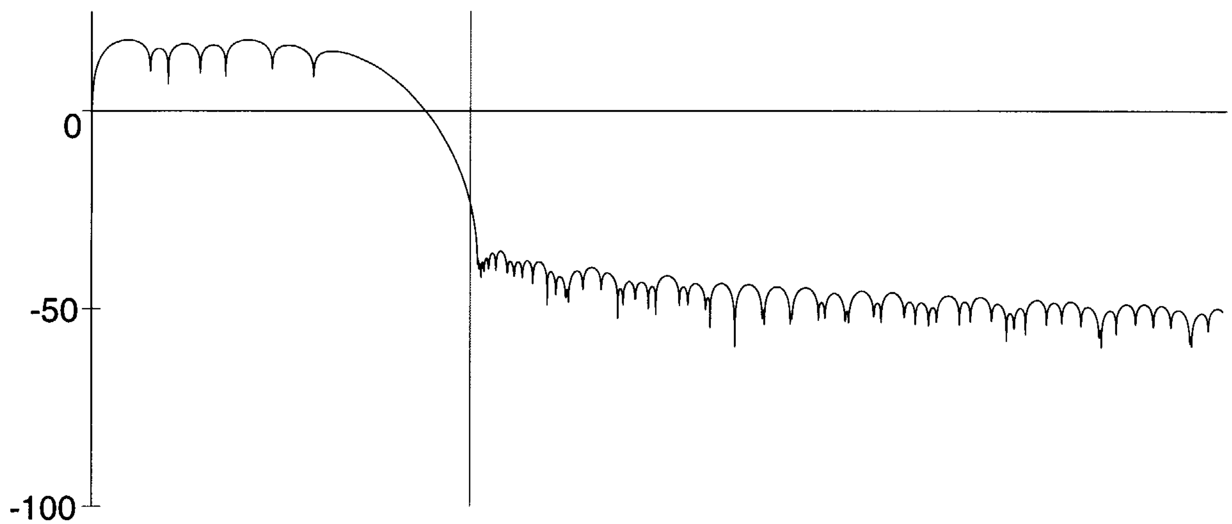
Multi-taper Methods

Average the first K Slepian tapers for $K < (2NW\Delta t - 1)$

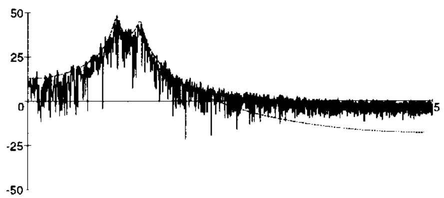
$$\hat{S}^{(N)}_{\text{mt}}(f) \equiv \frac{1}{K}\sum_{k=1}^{K} S^{(N)}_{s(k)}(f),$$

this is equivalent to using a kernel

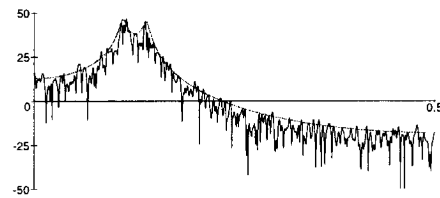
$$\mathcal{W}^{(N)}_{k}(f) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{W}^{(N)}_{k}(f).$$



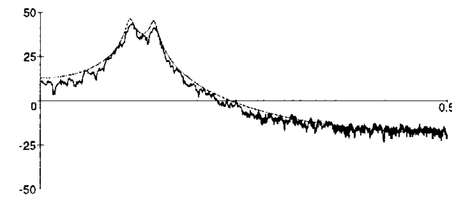
A graph of $10\ \log_{10}(\mathcal{W}^{(N)}_{(\text{mt})}(f))$ with $K = 4$.



periodogram



single taper



multi-taper

Harmonic Analysis using Multi-taper Methods

$X(t) = Ae^{i2\pi f_0 t} + N(t)$ to estimate A minimise the spectral estimator of $X(t) - \hat{A}e^{i}$

Using a windowed periodogram we have

$$\hat{S}_w^{(N)}(f_0) = \Delta t \left| J(f_0) - \tilde{w}^{(N)}(0)\hat{A}_w \right|^2,$$

where

$$J(f) \equiv \sum_{r=1}^{N} w_r X_r e^{-2\pi i f r \Delta t} \quad \text{and} \quad \tilde{w}^{(N)}(0) = \sum_{r=1}^{N} w_r$$

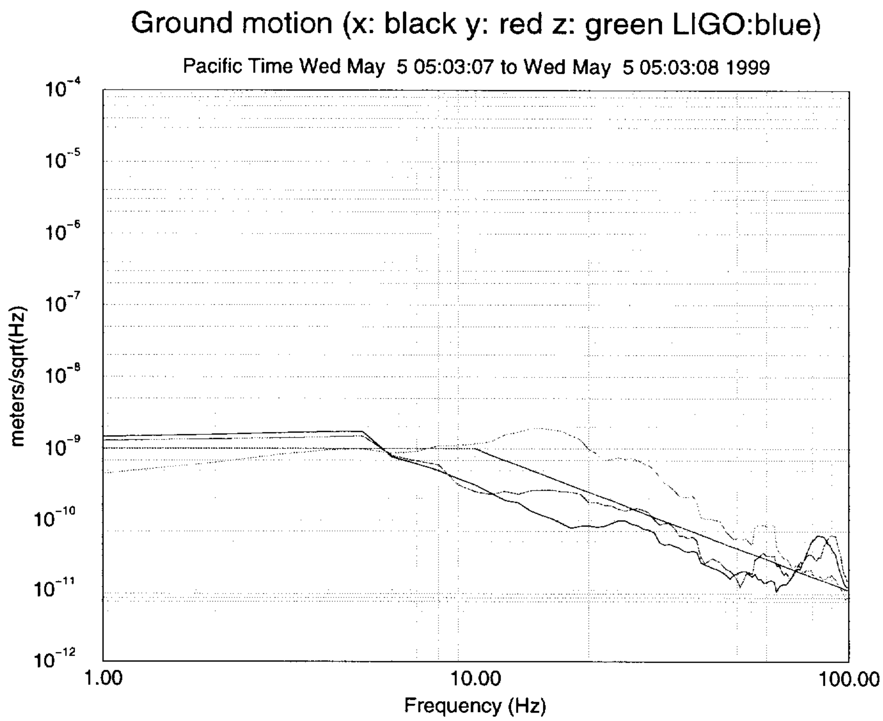
giving $\hat{A}_w = J(f_0)/\tilde{w}^{(N)}(0)$

For a multi-taper spectral estimate: $\hat{A}_{\mathrm{mt}} = \dfrac{\sum_{k=1}^{K} \tilde{w}_k^{(N)}(0) J_k(f_0)}{\sum_{k=1}^{K} \left[w_k^{(N)}(0) \right]^2},$

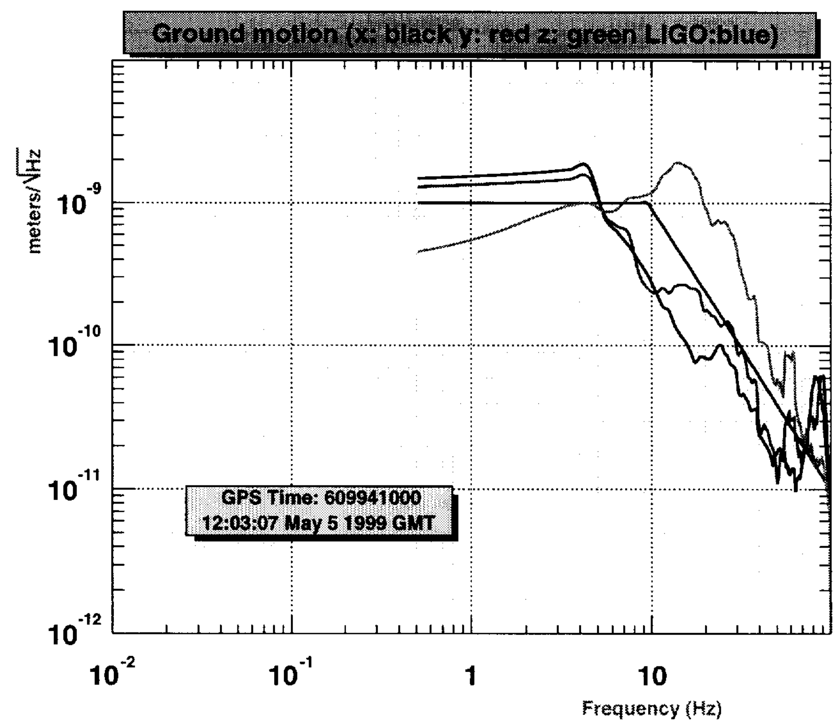
Also $F(f_0) = (K-1)\dfrac{|\hat{A}_{\mathrm{mt}}|^2}{\hat{S}_{\mathrm{mt}}^{(N)}(f_0)}$ is $F(2, 2K-2)$.

Implemented within GRASP by the program `remove_spectral_lines()`.

GRASP/xmgr output



DMT/Root output



```
int main(int argc,char **argv) {
     void graphout(float,float,int);
     float tstart=1.e35,srate=1.e-30,tmin,tmax,dt;
     float *dataf[3],freq,norm2;
     double time=0.0;
     int i,j,seq=0,code,npoint=NPOINT;
     short *data[3];
     struct fgetinput fgetinput;
     struct fgetoutput fgetoutput;

     /* number of channels */
     fgetinput.nchan=3;

     /* source of files */
     fgetinput.files=framefiles;

     /* storage for channel names, data locations, points returned, ratios */
     fgetinput.chnames=(char **)malloc(fgetinput.nchan*sizeof(char *));
     fgetinput.locations=(short **)malloc(fgetinput.nchan*sizeof(short *));
     fgetoutput.npoint=(int *)malloc(fgetinput.nchan*sizeof(int));
     fgetoutput.ratios=(int *)malloc(fgetinput.nchan*sizeof(int));

     /* set up channel names, etc. for different cases */
     fgetinput.chnames[0]="H0:PEM-LVEA_SEISX";   /* seismometer X */
     fgetinput.chnames[1]="H0:PEM-LVEA_SEISY";   /* seismometer Y */
     fgetinput.chnames[2]="H0:PEM-LVEA_SEISZ";   /* seismometer Z */
     fgetinput.inlock=0;

     /* number of points to get */
     fgetinput.npoint=npoint;

     /* don't seek, we need the sample values! */
     fgetinput.seek=0;

     /* but we don't need calibration information */
     fgetinput.calibrate=0;

     /* now loop ... */
          while (numtodo-- > 0) {
               /* seek, or get the sample values */
               code=fget_ch(&fgetoutput,&fgetinput);

               /* elapsed time, sample rate */
               tstart=fgetoutput.dt;
               srate=fgetoutput.srate;
```

1

GRASP code

```
DSeismic::DSeismic(int argc, const char* argv[])
  :DatEnv(argc, argv), nPoint(256), nChan(3), sRate(256.0) {

    //-------------------------------- Check file name
    const char* config;
    config = argv[argc-1];
    if (!config || !*config) {
      cerr<< "Seismic: configuration file not specified" << endl;
      return;
    }


    //-------------------------------- Get parameters.
    ifstream ifs(config, ios::in);
    if (!ifs) {
      cerr << "Seismic: Unable to open configuration file " << config << endl;
      return;
    }

    cout << "# Seismic data from config file: " << config << endl;


    for (int id=0 ; id<nChan ; id++) {
      ifs.getline(chanv[id].mName, sizeof(chanv[id].mName));
      chanv[id].mTS = new TSeries(Time(0), Interval(0.0), nPoint);
      chanv[id].mFS = new FSeries;
      In.addChannel(chanv[id].mName, 0, &(chanv[id].mTS));
      cout << "# Added channel: " << chanv[id].mName << endl;
    }

    ifs.close();


  //-------------------------------- Loop over sampling intervals

    In.setBuffer(0);      // Take up data as requested
```

1

DMT code

```
gROOT->Reset();
gStyle->SetOptStat(0);    //    Don't show statistics
gStyle->SetOptTitle(0);  //    Don't show individual tiles -- we'll add a global title

TCanvas *c = new TCanvas("c", "Ground motion (x: black y: red z: green LIGO:blue)",10,10,788,750);

c->SetHighLightColor(2);
c->SetFillColor(10); // White in the default palette - see TROOT::GetListOfColors
c->SetBorderSize(2);
c->SetLogx();
c->SetLogy();
c->SetGridx();
c->SetGridy();
c->SetTickx();
c->SetTicky();
c->SetLeftMargin(0.12);
c->SetRightMargin(0.0);
c->SetTopMargin(0.08);
c->SetBottomMargin(0.12);


TH1F *ligo  = new TH1F("ligo","LIGO standard",nActual,0.01,100);
TH1F *seisX = new TH1F("seisX","Ground motion x",nActual,0.01,100);
TH1F *seisY = new TH1F("seisY","Ground motion y",nActual,0.01,100);
TH1F *seisZ = new TH1F("seisZ","Ground motion z",nActual,0.01,100);

for (i=0;i<nActual;i++) {
  dy[0][i]=y[0][i];
  //  ligo->Fill(x[i],y[0][i]);
  seisX->Fill(x[i],y[1][i]);
  seisY->Fill(x[i],y[2][i]);
  seisZ->Fill(x[i],y[3][i]);
}
ligo->FillN(nActual,x,dy[0]);  // This only works for doubles - hence the contorsion

ligo->SetMinimum(1.e-12);  // Sey y_min
ligo->SetMaximum(1.e-8);   // Set y_max
ligo->SetLineColor(4);     // Blue in the default palette - see TROOT::GetListOfColors
ligo->SetLineWidth(2);
ligo->SetMarkerColor(10);
ligo->Draw("C**");           // Draw a continuous line between points

addFineGrid(xMin,xMax,yMin,yMax); //   Add a fine grid

ligo->Draw("CSAME");       // Redraw so graph appears above grid


seisX->SetLineColor(1);    // Black in the default palette - see TROOT::GetListOfColors
seisX->SetLineWidth(2);
```

1

```
        seisX->Draw("CSAME");      // Continuous line on the same graph
        seisY->SetLineColor(2);    // Red in the default palette - see TROOT::GetListOfColors
        seisY->SetLineWidth(2);
        seisY->Draw("CSAME");
        seisZ->SetLineColor(3);    // Green in the default palette - see TROOT::GetListOfColors
        seisZ->SetLineWidth(2);
        seisZ->Draw("CSAME");

        c->RedrawAxis();           //  Redraw so axes appears above lines

        TLatex *l1 = new TLatex(-2.45, -9., "meters/#sqrt(Hz)");  //  y-axis title
        l1->SetTextAlign(22);
        l1->SetTextSize(0.03);
        l1->SetTextAngle(90.);
        l1->SetTextColor(46);
        l1->Draw();

        TText *l2 = new TText(1.0, -12.5, "Frequency (Hz)"); //  x-axis title
        l2->SetTextAlign(22);
        l2->SetTextSize(0.03);
        l2->SetTextColor(46);
        l2->Draw();

        //  TPaveLabel *title = new TPaveLabel(0.2, 0.94, 0.85, 0.99,
//                                         "Ground motion (x: black y: red z: green LIGO:blue)");
//      title->SetFillColor(42);
//      title->Draw();

        TPaveText *pt = new TPaveText(0.2, 0.26, 0.45, 0.325,"NDC");
        pt->SetFillColor(18);
        sprintf(filename," GPS Time: %d",gpsStartTime);
        // time2=gpsStartTime-7*3600;   7 hours - Hanford correction ?
        TText *text = pt->AddText(filename);
        TimeStr(ts->getStartTime(), UTCdate,"%H:%N:%S %M %d %Y %Z");
        text = pt->AddText(UTCdate);
        pt->Draw();

        c->Update();
```

2

DMT/ROOT graphics code

Note 1, Linda Turner, 05/09/00 09:55:30 AM
LIGO-G000086-00-D